AMS Procurement Desktop-Defense™

Troubleshooting Transaction Log Errors*

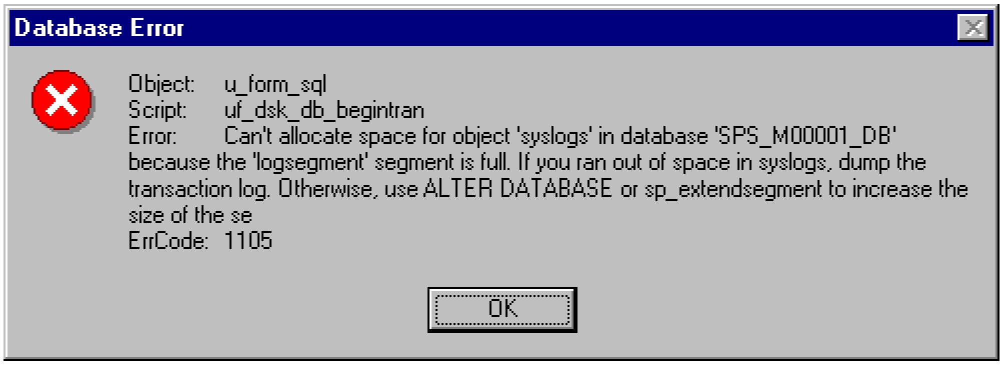
March 23, 2000

* Because this topic is not covered by the current PMO funded SPS Helpdesk Agreement, this document has been provided to help you resolve this issue. If you still need assistance after reviewing this document, please contact a representative from your Customer Support Team.

Table of Contents

1. Problem

A user is working in PD^2 and the following error appears.



```
Object:   u_form_sql
Script:   uf_dsk_db_begintran
Error:    Can't allocate space for object 'syslogs' in database 'SPS_M00001_DB'
because the 'logsegment' segment is full. If you ran out of space in syslogs, dump the
transaction log. Otherwise, use ALTER DATABASE or sp_extendsegment to increase the
size of the se
ErrCode:  1105
```

Figure 1: Error 1105

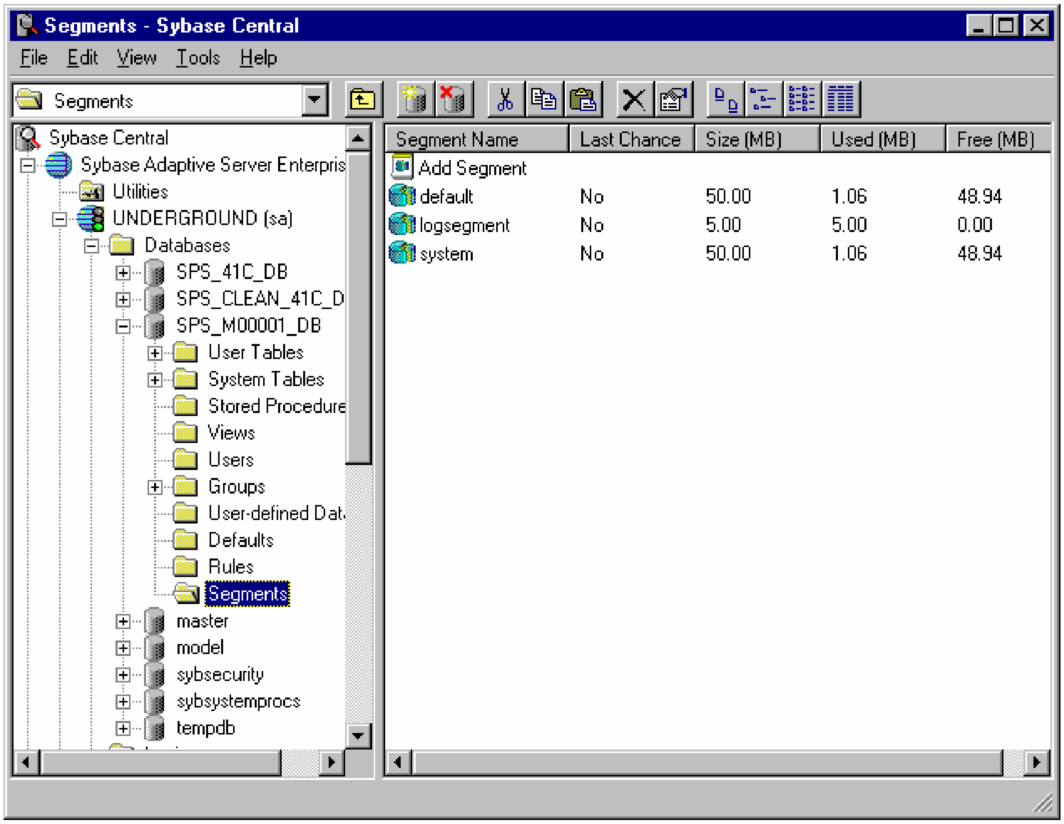
Error: "Can't allocate space for object ***'syslogs'*** in database '*<database_name>*' because the '*logsegment*' segment is full. If you ran out of space in ***syslogs***, dump the transaction log. Otherwise, use ALTER DATABASE or sp_extendsegment to increase size of the segment."
ErrCode: 1105"

The user ran out of space in the transaction log on the database.

2. Verification

2.1 Using Sybase Central

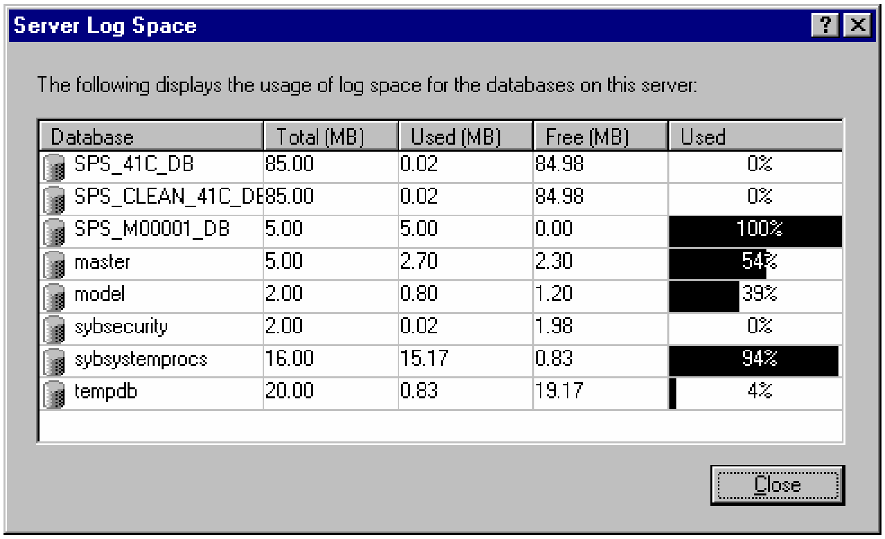
Connect to the server using Sybase Central. After logging in as ‘sa’, double click on the folder named "Databases". Locate the database that is listed in the error message and double-click on it. When the list of folders appears select the “segments” folder (See Figure 2). The segments folder lists the three segments that make up the database. Those segments are default, logsegment and system segment. The logsegment is used to store transaction log information. If your database is set to "truncate log on checkpoint" then this segment should not fill up under normal circumstances. (However, in this example this option has been turned off.) Refer back to the error message and verify which segment is full. If the error mentioned the logsegment then there would be a 0.00 (or close to 0.00) in the column under Free (MB).



Segment Name	Last Chance	Size (MB)	Used (MB)	Free (MB)
Add Segment				
default	No	50.00	1.06	48.94
logsegment	No	5.00	5.00	0.00
system	No	50.00	1.06	48.94

Figure 2: Segments Folder

There is a second method for verifying available log space using Sybase Central. After logging into the server right click on the server name and select “Log Space” from the pop-up menu. Or you can highlight the server name and select “Log Space” from the File menu at the top of the window. A Server Log Space window will appear listing the current size log and percent used for each database on the server (See Figure 3). Locate the database that is listed in the error message. If the logsegment for that database is full then the percent used column will read 100% (or close to 100%).



The following displays the usage of log space for the databases on this server:

Database	Total (MB)	Used (MB)	Free (MB)	Used
SPS_41C_DB	85.00	0.02	84.98	0%
SPS_CLEAN_41C_DE	85.00	0.02	84.98	0%
SPS_M00001_DB	5.00	5.00	0.00	100%
master	5.00	2.70	2.30	54%
model	2.00	0.80	1.20	39%
sybsecurity	2.00	0.02	1.98	0%
sybsystemprocs	16.00	15.17	0.83	94%
tempdb	20.00	0.83	19.17	4%

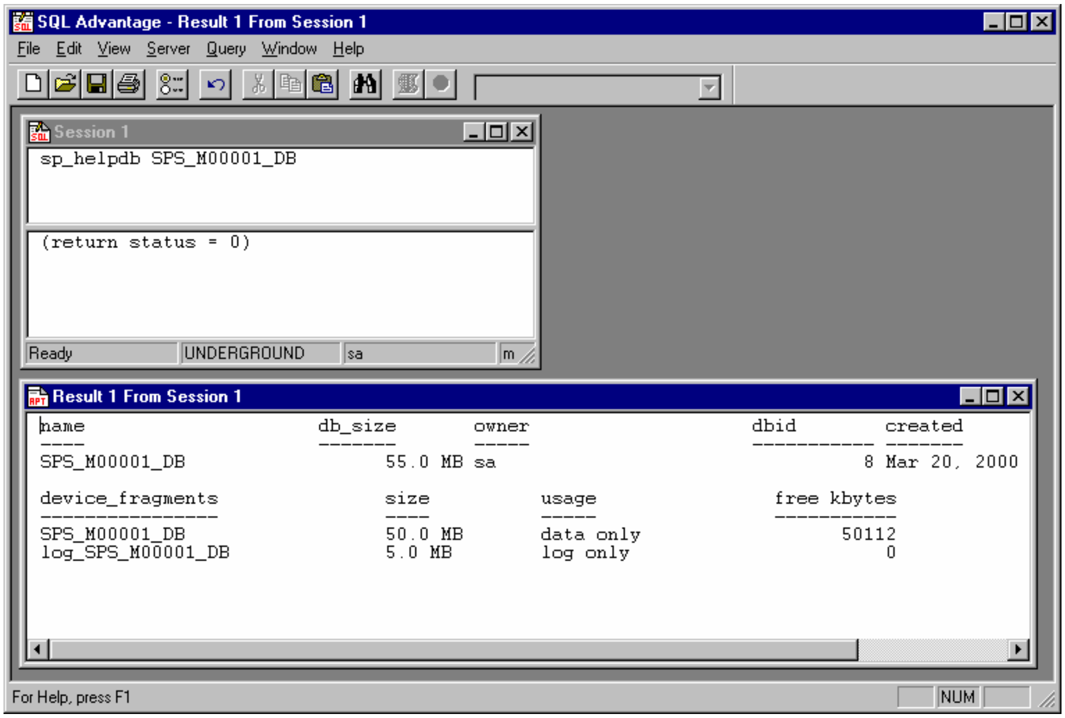
Figure 3: Log Space Window

2.2 Using SQL Advantage

In WISQL or SQL Advantage execute the following command.

sp_helpdb ***<dbname>***

<dbname> = the name of the database which has a full transaction log.



```
sp_helpdb SPS_M00001_DB
```

```
(return status = 0)
```

```
name                          db_size       owner                        dbid          created
----                          --------      -----                        ------------  -------
SPS_M00001_DB                    55.0 MB sa                                         8 Mar 20, 2000

device_fragments                 size          usage            free kbytes
----------------                 ----          -----            -----------
SPS_M00001_DB                    50.0 MB       data only              50112
log_SPS_M00001_DB                5.0 MB        log only                   0
```

Figure 4: Results from sp_helpdb

When the results come back note the information under column entitled “free kbytes”. This number will be 0 (or close to 0) if the logsegment is full.

3. Solution

3.1 Dumping the Transaction Log

Whenever a command is issued that inserts, updates or deletes a row in a table, the transaction is also sent to the transaction log. As each transaction is committed to the database the amount of space in the log is reduced. If your site backs up its transaction data dumping the transaction log frees up space by writing committed transactions to disk and then removing them from the log. When the transaction log fills up, it must be dumped or truncated before anyone can continue working in the system.

There are four ways to dump the transaction log.

- Dump transactions to a dump device. This stores the committed transactions to a file that can be used in conjunction with the last database backup to make a full recovery of the database.
- Dump transactions with truncate_only. This will truncate the transaction log without copying the committed transactions to a file.
- Dump transactions with no_log. This will truncate the log without recording the event. This should only be done when the other dump transaction commands fail because of insufficient log space.
- Reboot the server. This will terminate all uncommitted transactions and allow you to truncate the log. Do this when all other attempts to dump the transaction log fail.

Note: When the “truncate log on checkpoint” option is on, you cannot dump the transaction log to a dump device because changes to your data are not recoverable from transaction log dumps. In this situation, issuing the “dump transaction...to” command produces an error message instructing you to use dump database instead.

3.1.1 Dumping the Transaction Log to a Dump Device

If you have created dump devices for backing up the transaction log then execute the following command in WISQL or SQL Advantage to dump your transaction log.

```
dump tran <database_name> to <device_name>
```

<database_name> = the name of the database that is associated with log that needs to be dumped.
<device_name> = the name of your dump device.

Example

```
dump tran SPS_M00001_DB to TRANS_BACKUP
go
```

Execute the following command in WISQL or SQL Advantage if you want to use a physical file to backup up your transaction log.

```
dump tran <database_name> to "<physical_file>"
```

<database_name> = the name of the database that is associated with log that needs to be dumped.
<physical_file> = the directory and file name of the .dmp file.

Example

```
dump tran SPS_M00001_DB to
"c:\sybase\backup\TRANS_BACKUP.dmp"
go
```

Note: If the physical file exists, the previous backup will be over written. If the physical file does not exist then one will be created.

Note: This command will not work if the "truncate log on checkpoint" option is turned on in the database.

3.1.2 Dumping the Transaction Log with Truncate Only

If you are not currently keeping backups of your transaction log and you do not need to retain this information, then you can execute the following command in WISQL or SQL Advantage to dump your transaction log without making a copy of it. This will remove committed transaction from the log while retaining uncommitted transactions.

```
dump tran <database_name> with truncate_only
```

<database_name> = the name of the database that is associated with log that needs to be dumped.

Example

```
dump tran SPS_M00001_DB with truncate_only
go
```

Note: This command will not work if there is no space remaining in the transaction log because it requires a small amount of space in the transaction log in order to run.

3.1.3 Dumping the Transaction Log with No Log

If your Transaction Log is completely full (i.e. space remaining is 0.00MB) then you will not be able to dump the transaction log using any of the previously described methods. At this point you must execute the following command in WISQL or SQL Advantage to dump your transaction log. The following command will not store the data in the transaction log to a file and the "dump tran" action will not be recorded in the error log.

```
dump tran <database_name> with no_log
```

<database_name> = the name of the database that is associated with log that needs to be dumped.

Example

```
dump tran SPS_M00001_DB with no_log
go
```

Note: Sybase recommends that you backup your database immediately after performing this function.

3.1.4 Rebooting the Server

If all other attempts to dump the transaction log fail, then you can reboot the Sybase server to terminate all uncommitted transactions. Once the server has restarted, log back into SQL Advantage and use one of the methods identified in Section 2: Verification to check the current space available in the log. If the log is still full then use one of the methods described above to dump the transaction log.

3.2 Setting Database Options

After you have successfully dumped the transaction log, take a few moments to verify the current database settings. As stated in Section 2: Verification, the transaction log will not fill up if the "truncate log on checkpoint" option is set on the database.

Note: For sites that perform a routine dump of their transaction log this option will not be set. However, for other sites that are not backing up their transaction logs, AMS recommends that this option be set in order to avoid this problem.

In order to verify whether the "truncate log on checkpoint" option is set execute the following command in WISQL or SQL Advantage.

```
sp_helpdb
```

Locate the name of your database and scroll over to the status column. In most cases the option will read "select into/bulkcopy, trunc log on chkpt". If these options are not set then follow these instructions to set the options.

To change a database option in a PD² database you must execute the sp_dboption procedure from the master database.

The syntax for sp_dboption is as follows:

```
sp_dboption <dbname>, "<optname>", {true | false}
```

<dbname> = the name of the database for which you are setting the option.
<optname> = the name of the option that you want to set.
<true|false> = the choice of setting. True = on. False = off.

Example

```
sp_dboption SPS_M00001_DB, "trunc log", true
go
sp_dboption SPS_M00001_DB, "select into", true
go
```

After setting the options run the *checkpoint* command in your database for the changes to take effect.

```
use <dbname>
go

checkpoint
go
```

<dbname> = the name of the database for which you have changed the option.

3.3 Increasing the Size of Your Transaction Log

In some cases you may need to increase the size of your transaction log to prevent it from filling up. For upgrade purposes, AMS recommends that the size of your transaction log be as big the largest (non-text/image-containing) table in the database. But in most cases the transaction log is installed with 100MB of space.

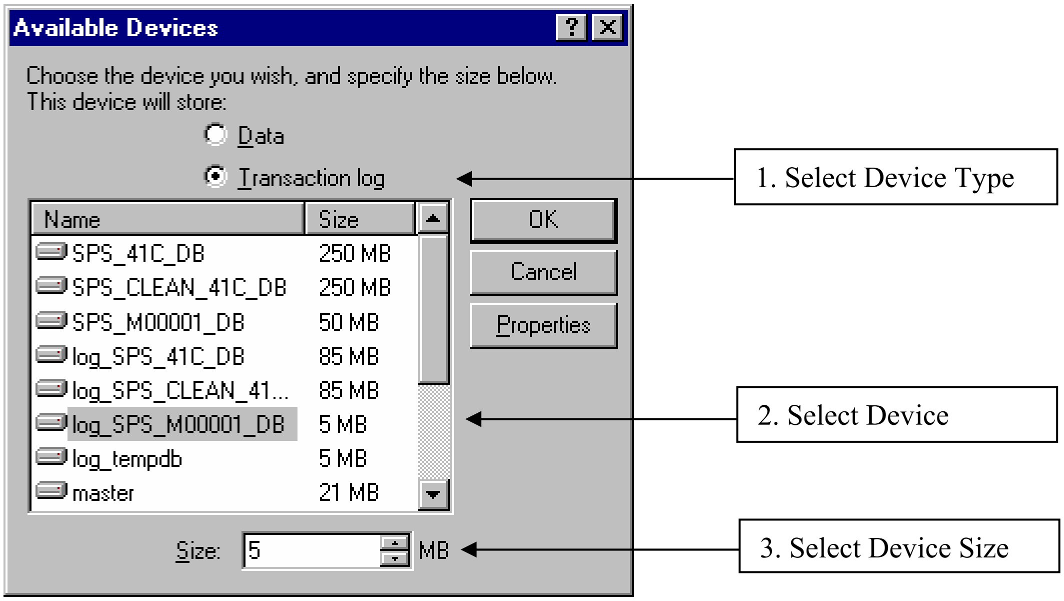
If you need to increase the size of your transaction log or add a transaction log to a database that does not have one then follow these steps.

1. Create a database device[1]. Be sure to use the proper naming convention when creating a device for transaction log data. Example: **log_<*database_name*>_ADD1**. This identifies the device as the first addition to the log for the database.

2. Attach the data device to the database[1]. Attaching a device for storing transaction log information differs slightly from attaching a device for storing data. Refer to the following sections for these changes.

3.3.1 *Using Sybase Central*

When you arrive at the screen where you are prompted to select an available device, be sure to select the “Transaction Log” radio button as your device type.



Available Devices

Choose the device you wish, and specify the size below.
This device will store:

○ Data

● Transaction log

Name	Size
SPS_41C_DB	250 MB
SPS_CLEAN_41C_DB	250 MB
SPS_M00001_DB	50 MB
log_SPS_41C_DB	85 MB
log_SPS_CLEAN_41...	85 MB
log_SPS_M00001_DB	5 MB
log_tempdb	5 MB
master	21 MB

Size: 5 MB

1. Select Device Type

2. Select Device

3. Select Device Size

Figure 5: Add Device Window

3.3.2 *Using WISQL or SQL Advantage*

You can attach a transaction log or increase the size of an existing transaction log on a database by executing the following command in WISQL or SQL Advantage.

```
alter database <dbname> log on <devname> = <size>
```

<dbname> = the name of the database.
<devname> = the name of the new log device that you created.
<size> = the size of the device in MB.

Example

```
alter database SPS_M00001_DB log on
log_SPS_M00001_DB_ADD1 = 100
go
```

[1] For detailed instructions on creating a database device and attaching it to the database follow the instructions in the document entitled “How to Increase the Size of Your PD^{2} Database”.